저전력 회로를 위한 비트 단위의 연산 최적화

엄준형 김태환
한국과학기술원 전산학과
첨단정보기술 연구센터(AITrc)
jhum@vlsisyn.kaist.ac.kr tkim@cs.kaist.ac.kr

A Bit-level Arithmetic Optimization for Low-Power Circuits

Junhyung Um Taewhan Kim
Dept. of EECS, Korea Advanced Institute of Science and Technology
and Advanced Information Technology Research Center(AITrc)

요 약

고속 회로 합성에 있어서, Wallace 트리 스타일은 연산을 위한 가장 효율적인 수행 방식의 하나로 인식 되어졌다. 그러나, 이러한 방법은 빠른 곱셈기의 수행이나 여러가지 연산수행에 있어, 입력 시그널을 고려하지 않은 일반적인 구조로 수행되어졌다. 본 논문은 연산기에 있어서 이러한 제한점을 극복하는 문제를 다룬다. 우리는 캐리-세이브 방법을 덧셈, 뺄셈, 곱셈이 혼합되어 있는 일반적인 연산 회로에 적용한다. 그 결과 효율적인 회로를 생성하며, 시그널들의 임의의 시그널 스위칭 변화에 대해 회로의 전력 소모를 최적화 한다. 우리는 이러한 최적화 방법을 여러 디지털 필터에 적용시켜 보았고 이는 기존의 비트 단위가 아닌 캐리-세이브 수행방법보다 상당한 양의 전력 소모의 향상을 보였다.

1 서론

상위 단계의 합성에 있어, 연산에 대한 전력 소모/면적/지연시간의 범위가 주어진다면, 연산의 적절한 할당은 전력 소모[1]을 상당히 감소하는 회로를 만들 수 있다. 데이터 경로의 switching capacitance는 연산이 어떤 모듈에 할당되는가, 그리고 변수들이 어떤 레지스터에 할당되는가에 상당히 많은 부분을 의존한다. RTL 합성에서는, 회로의 전력 소모를 측정하고 분석하는것이 상위 단계의 합성에서 고려되어야 하는 중요한 연구 분야이다. 논리 합성 단계에선 여러가지 최적화 방법, 예를 들어 경로 balancing, 클럭 gating, encoding, 그리고 retiming등이 switching activity를 줄이기 위한 방법으로 연구되어 왔다.

연산 회로에 있어 덧셈은 상당히 자주 행해지는 연산이기 때문에 저전력의 덧셈 연산의 수행은 매우 중요한 문제이다. 각 피연산자 X_i가 n_i 비트인 연산식 $F = X_1 + X_2 + \cdots + X_m$ 를 실현하는 회로를 만드는데 있어 가장 효율적이고 흔히 사용되는 방법은 캐리-세이브-가산기(CSA)를[2] 이용하는 것이다. 여기서 주목할 점은 CSA 수행은 단지 덧셈에만 제한되지 않는다는 점이다. 우리는 뺄셈(예를 들어, $x - y = x + \bar{y} + 1$)이나 곱셈같은 다른 연산도 덧셈으로 변환할 수 있다. 그림 1은 캐리-세이브 가산기를 수행하는 두 단계의 구조를 보여준다. 첫번째 단계에서는 그림 2에서 보이는 full-adder를 사용한 캐리-세이브 가산을 수행한다. 이때 full-adder(FA)간에 캐리-지연이 존재하지 않는다. 그리고, 두번째 단계에서는 첫번째 단계에서 생성된 두개의 피연산자를 캐리-지연 연산(carry-propagation addition) 을 이용하여 마지막 결과를 생성한다.

이러한 피연산자의 덧셈 수행은 비트 단위의 addend 행렬로 나타내어진다. 예를 들어, $X = x_3x_2x_1x_0, Y = y_3y_2y_1y_0, Z = z_2z_1z_0$, 그리고 $W = w_2w_1w_0$ 일때 연산식 $F = X + Y + Z + W$에 대한 행렬은 그림 3에서 볼 수 있다. 최적화 문제는 FA를 할당함으로써 각 열에 많아야 두개의 시그널들만 남도록 addend 행렬을 변환하는 것이다.

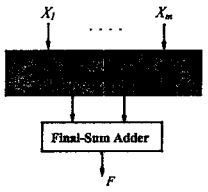

그림 1: 비트 단위의 캐리-세이브 최적화 문제

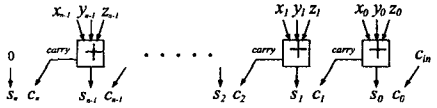

그림 2: n 비트의 캐리-세이브 가산기

우리는 그림 1의 첫번째 단계에서 보여지는 이러한 방법을 *FA 트리 할당*이라 부르고, 두번째 단계의 수행을 *Final-adder 할당*이라 부른다. 여기서 최적화 문제는, 주어진 연산식 F에 대한 addend 행렬과 함수적으로 동치 관계에 있는 가장 적은 전력 소모를 가지는 FA 트리를 만드는 것이다.

col-3	col-2	col-1	col-0
x_3	x_2	x_1	x_0
y_3	y_2	y_1	y_0
	z_2	z_1	z_0
	w_2	w_1	w_0

그림 3: 초기 addend 행렬의 예

2 전력 소모 모델과 문제 정의

CMOS 게이트에서, 대부분의 전력 소모는 출력이 charge되거나 discharge되는 출력 트랜지션이 일어날때 생긴다. 우리는 W_s와 W_c를 transition이 발생한 경우 FA의 sum out s와 carryout c에서 일어나는 전력 소모를 나타내기 위한 상수로 각각 사용한다.

우리는 연속된 입력 시그널과 회로의 내부 노드를 나타내기 위해 확률적 모델을 사용한다. 우리는 x의 값이 1일 확률을 나타내기 위해 random variable $p(x)$를 사용한다. 우리는, glitch로 인한 시그널 transition을 무시하기 위해 zero-gate 지연시간 모델을 사용한다. 그러면, 시그널 x에 대한 평균적인 switching activity는 $E_{switching}(x) = p(x)\cdot(1-p(x))$로 나타낼 수 있다. 결과적으로 우리가 해결하고자 하는 FA 트리 할당 문제는 주어진 연산식에 대해 $E_{switching}(x)$를 최소화 시키는 FA-트리 T를 찾아내는것, 즉, F의 전력 소모량을 최소화 시키는것이다.

$$E_{switching}(T) = \sum_{v\in V(T)} \{W_s\cdot p(v_s)(1-p(v_s)) + W_c\cdot p(v_c)(1-p(v_c))\}$$

$V(T)$는 T안의 FA들의 집합이고, v_s와 v_c는 각각 $V(T)$의 sum과 carryout 시그널을 나타낸다. 그러면, 우리의 최적화 문제는 다음과 같이 나타낼 수 있다:

문제: *다음과 같이 주어진 연산식*

$$F = X_1 + X_2 + \cdots + X_m \quad (1)$$

에 대해, $E_{switching}(T)$를 최소로 하는 FA-트리를 찾는다. 이때, X_i $(i=1,2\cdots m)$는 n_i-비트 $(=x_{i,n_i-1}\cdots x_{i,1}\,x_{i,0})$시그널이며, 각 시그널 확률은 $p(x_{i,j})$, $j=1,2,\cdots,n_i$이다.

먼저, 식 1의 간단한 경우부터 생각해 보자.

$$F = X_1 + X_2 + \cdots + X_m \quad (2)$$

$(bit\_width(X_i) = 1, i = 1,2\cdots m)$.

위의 문제에 대한 FA-트리 할당은 FA의 sum과 carryout 시그널 확률에 의존한다. 그리고, FA의 세개의 입력 시그널 x, y, z와 각 시그널 확률 $p(x)$, $p(y)$, $p(z)$에 대해 $q(v) = p(v) - 0.5$로 놓으면, $q(s)$와 $q(c)$는 다음과 같은 값을 갖는다.

$$\begin{aligned} q(s) &= 4\cdot q(x)\cdot q(y)\cdot q(z), \\ q(c) &= 0.5\cdot(q(x)+q(y)+q(z)) - 2\cdot q(x)\cdot q(y)\cdot q(z) \end{aligned} \quad (3)$$

이때, $p(v)(1-p(v)) = -(q(v))^2 + 0.25$이므로, $\Sigma p(v)(1-p(v))$를 최소화 하는것은 $-\Sigma(q(v))^2$를 최소화 하는것과 같다.

우리는 위와 같은 문제를 해결할때에, Final Adder는 서로 다른 지연시간과 회로 면적을 가지는 여러가지의 연산기로 구현될 수 있기 때문에, 우리는 문제 1을 조금 변형시켜 모든 입력으로부터 final adder의 입력이 되는 시그널까지의 전력 소모를 최적화 한다.

3 전력 소모를 고려한 FA-트리 할당을 위한 알고리즘

그림 4는 $E_{switching}(T)$의 값을 줄이는데에 있어, FA의 입력의 선택이 어떠한 영향을 주는지를 보이는 예이다. 우리는 이 예에서 W_c=W_s=1임을 가정하였다. 그림 4는 4개의 싱글 비트 x_1, x_2, x_3, x_4에 대해 서로 다른 FA-트리 T_1, T_2를 보인다. 이때, 각 x_i의 시그널은 $p(x_i)$로 주어져 있다$(q(x_i) = p(x_i) - 0.5)$. T_1과 T_2는 FA의 입력으로 서로 다른 시그널을 선택하였기 때문에, 서로 다른 switching activity의 합이 구해진다. 즉,

$$\begin{aligned} &E_{switching}(T_1) \\ &\quad = W_s\cdot\{-(q(s_1))^2 + 0.25\} + W_c\cdot\{-(q(c_1))^2 + 0.25\} \\ &\quad = -(q(s_1))^2 - (q(c_1))^2 + 0.5 = -0.089 + 0.5 = 0.411. \\ &E_{switching}(T_2) \\ &\quad = -(q(s_2))^2 - (q(c_2))^2 + 0.5 = -0.100 + 0.5 = 0.400. \end{aligned}$$

이는 그림 4(b)의 트리 T_2가 그림 4(a)의 트리 T_1보다 적은 양의 전력을 소모한다는 것을 의미한다. 명백하게 FA의 서로 다른 입력의 선택은 $E_{switching}(T)$에 영향을 미치는 내부 구조를 바꾸며, 이러한 전력 소모를 측정하는 값은 sum 시그널을 위한 값(*i.e.*, $q(s)$)과 캐리 시그널을 위한 값(i.e., $q(c)$)의 weighted 합이다. 수식(4)의 $q(s)$와 $q(c)$에서, 우리는 다음과 같은 성질을 발견 할 수 있다.

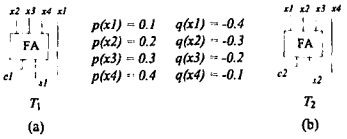

그림 4: FA의 시그널 선택에 따른 전력 소모의 차이를 보이는 예

관찰: sum 시그널의 스위칭으로 발생되는 전력 소모를 최소화하기 위해 우리는 $(q(s))^2$, $\{4q(x)\cdot q(y)\cdot q(z)\}^2$를 최대화 하여야 한다. 이를 이루기 위해, *FA의 입력으로 가장 큰 $|q|$값을 갖는 세개의 입력을 선택하여야 한다.* 또한, 이러한 선택은 같은 FA의 값인 $(q(c))^2$를 크게 할 수 있다. 예를 들어,

$$(q(c))^2 = \{0.5(q(x)+q(y)+q(z)) - 2q(x)\cdot q(y)\cdot q(z)\}^2$$

는 $q(x), q(y), q(z)$가 모두 양수이거나 음수일때 위와 같은 선택이 가장 큰 값을 갖는다.

위와 같은 관찰을 바탕으로, 우리는 수식 (2)(하나의 열에 대한 할당)을 실행하는 알고리즘을 다음과 같이 제시한다.

알고리즘*SC_LP*은 각 반복에서 statement_B을 따라 세 개의 입력을 선택한다. *SC_LP*의 M_0에서 논리값 0을 포함하는 것은 해당 비트열이 반복적인 수행의 마지막에 HA를 할당시키기 위함이다. FA-트리 T에 대해, $P_{switching}(T) = P_{switching}(T) - \sum_{v\in V(T)}\{\sqrt{W_s}\cdot|q(v_s)| + \sqrt{W_c}\cdot|q(v_c)|\}$ 와 $E_{switching}(T)$는 밀접히 관련되어 있다. 즉, $P_{switching}(T)$의 값은 $E_{switching}(T)$가 감소함에 따라 감소한다. 성질 1은 특정한 조건에서, *SC_LP*가 P를 최소화 하는 FA-트리를 생성함을 증명한다:

성질 1 *수식 (2)의 주어진 입력행렬에 대해, SC_LP는 $2\cdot\sqrt{W_s} \geq \sqrt{W_c}$이고 입력 시그널의 확률이 모두 0에서 0.5사이이거나, 아니면 모두 0.5에서 1 사이 일때, $P_{switching}(T)$를 최소화 하는 FT-트리를 생성한다.*

알고리즘 *SC_LP*(M_0):
한 비트 열에 대한 FA-트리 할당 (식 (2)의 F)
• $M_0 = \{x_{i,0}\}$, $1 \le i \le m$ /* 해당 열의 입력 */
• 만약 $|M_0|$ 이 홀수이면, logic value 0을 (called x') M_0에 포함 /* HA를 할당하기 위함 */
while ($|M_0| \ge 3$) {
• M_0에서 $|p(x) - 0.5|$의 값을 가장 크게 하는 *세개의* 입력 선택(=$|q(x)|$); (statement_a)
• 새로운 FA를 만들고 선택된 입력을 할당;
(선택된 입력들 중 x'가 포함될 경우 HA를 만든다)
• M_0로부터 이미 할당된 세개의 입력을 없앤다;
• FA/HA의 sum port로부터의 입력을 M_0에 추가한다;
}

성질 2는 우리가 sum 시그널만을 고려하였을때, 제시된 알고리즘 $SC|_LP$가 스위칭으로 인한 전력 소모를 최소화 하는 FA-트리를 생성함을 증명하고 있다.

성질 2 *식 (2)에 주어진 입력 행렬에 대해, 알고리즘 SC_LP는 $W_c = 0$인 경우 $E_{switching}(T)$를 최소화 하는 FA-트리를 생성한다.*

그러나, 일반적인 W_s, W_c의 값과 시그널들의 분포에 대해 *SC_LP*는 최적의 해를 보장하진 않는다. 왜냐하면, 이는 해당열의 FA에서 생성되는 carryout들이 다음의 열에 미치는 영향이 상당히 복잡하기 때문이다. 또한, 우리의 최적화 알고리즘은 "FA 분해"를 하는것으로 기존의 primitive 게이트 분해, 즉 AND-gate[3], XOR-gate[4]와는 다르다. AND, XOR 게이트등은 분석이 비교적 용이한 반면에 FA는 좀더 복잡하며, 하나의 출력의 스위칭을 고려하는것과 두개의 출력의 스위칭을 동시에 고려하는것은 판이하게 다르기 때문이다. 그러나, 이전에 제시된 관찰과 성질 3은 우리의 *SC_LP*알고리즘이 비교적 적은양의 전력 소모를 가지는 FA-트리를 생성함을 보이며, 또한 우리의 알고리즘에 의해 생성된 캐리 아웃의 스위칭 activit가 아주 큰 양의 전력 소모를 가지고 오지는 않는것을 뒷받침한다.

성질 3 *하나의 입력행렬의 원소가 하나만 남을때, 생성된 모든 FA의 시그널 probability의 합은 FA가 어떠한 방법으로 할당되건간에 항상 일정하다.*

마지막으로, 제시된 관찰과 Properties 1, 2, 3를 기반으로 하여 우리는 알고리즘 *SC_LP*을 확장하여 여러개의 비트열 입력 행렬을 위한 저전력 FA-할당 알고리즘을 다룬다. 이러한 확장 알고리즘 *FA_ALP* (*Low Power를 위한 FA-tree 할당 알고리즘*)은, *SC_LP*을 부분 routime으로 이용하여 가장 오른쪽 비트열부터 왼쪽 비트열로 차례로 수행한다.

알고리즘 **FA_ALP(F):**
수식(1)의 F를 위한 최적 전력소모의 FA 트리 할당 :
• $n = max\{\, n_k \mid 1 \le k \le m \,\}$ /* n_k: *bit-width*(X_k) */
• $M_j = \{x_{i,j}\}$, $0 \le j \le n-1$
/* addend 행렬 M의 j번째 열의 시그널들의 집합 */
• $j = 0$; /* 가장 오른쪽의 열(최소 weight을 갖는 열) */
repeat {
• *SC_LP*(M_j)를 부른다;
• FA(만약 HA가 만들어졌다면 HA)에 의해 생성된 모든 시그널을 M_{j+1}에 넣는다:
(만약 M_{j+1}가 존재하지 않는다면, M_{j+1}를 새로 만들고, 생성된 시그널을 넣는다.)
• $j = j + 1$;
} **until** ($|M_s| \le 2$, $s = 1, 2, \cdots$)
/* 여기서, M 는 두개의 행만을 갖는다 */
• 두 행의 비트 width와 동일한 width를 갖는 final adder를 만든다;
• 두개의 피연산자를 생성된 final adder의 입력으로 할당한다;

4 실험

우리는 두가지의 FA-트리 할당 알고리즘을 수행하였다. 하나의 알고리즘, 즉, FA_{random}은 FA의 입력으로 할당되는 시그널을 *FA_ALP*을 수행하였다. 우리는 FA-트리 내부의 FA들의 출력의 시그널의 변화로 생성되는 전력 소모를 측정하였고 ($E_{switching}(T)$) 그 결과를 표 1에 정리하였다. 우리는 회로의 스위칭 확률로을 위해 random한 스위칭 확률을 이용하였으며, Synopsys *Design Power*[5]를 하나의 시그널이 변화할때 생성되는 전력 소모량를 나타내는 상수, 즉W_s와 W_c를 측정하기 위해 사용하였다. 이때, target library는 *lcbg10pv*(0.35*u*)이며, global operating voltage는 3.3이다. 이러한 비교는 *FA_ALP*에 제시된 시그널의 선택이 FA트리의 전력 소모를 감소시키는데 있어서 상당히 효율적임을 보이고 있다. 또한, 이러한 전력 소모는 상당히 안정적이어서, 우리의 알고리즘으로 FA-트리를 생성하는것이 위험성을 상당히 낮추는 결과를 가지고 오는것을 알 수 있다.

Design	FA_{random}	*FA_ALP*	Impr.
IIR	257 mW	240 mW	6.6%
Kalman	316 mW	281 mW	11.0%
IDCT	1406 mW	1324 mW	5.8%
Complx	330 mW	299 mW	6.6%
Serial-Adapter	324 mW	240 mW	25.9%
Average			11.8%

표 1: 회로의 전력 소모 비교

5 결론

본 논문에서, 우리는 비트 단위의 캐리-세이브 가산에 기초한 새로운 연산 회로 합성 알고리즘을 제시하였다. 곱셈 연산기에 대한 기존의 비트 연산 변환의 적용과는 다르게 우리는 덧셈, 뺄셈, 곱셈이 혼합된 일반적인 연산식에 이러한 방법을 적용하였으며, 각 입력 시그널들에 대한 동일한 스위칭 확률을 가정한 기존의 방법과는 다르게 일반적인 시그널에 대해 전력 소모를 줄이는 비트 압축 알고리즘을 제시하였다. 실험에서, 우리는 우리의 알고리즘이 비트 단계의 캐리 세이브 가산을 전반적인 가산 회로에 효율적으로 적용함을 보였으며, 이는 회로 전력 소모 에 상당한 향상을 가지고 오는것을 보였다.

감사의 글 : 본 논문은 첨단정보기술 연구센터(AITrc)를 통하여 과학재단의 지원을 받았음.

참조 서적

[1] A. P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Broderson, "Optimizing Power using Transformations", *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, Vol. 14, No. 1, pp. 12-31, January 1995.

[2] D. D. Gajski, "Parallel Compressors," *IEEE Transactions on Computers*, Vol. C-29, No.5, pp. 393–398, May 1980.

[3] H. Zhou and D.F. Wong, "An Exact Gate Decomposition Algorithm for Low-Power Technology Mapping", *Proc. of International Conference on Computer-Aided Design*, pp. 575-580, 1997.

[4] Unni Narayanan and C.L. Liu, "Low Power Logic-Synthesis for XOR Based Circuits", *Proc. of International Conference on Computer-Aided Design*, pp. 570-574, 1997.

[5] Synopsys Inc., *Power Compiler Reference Manual*, 1998.